

第2章 濃度変換

発表グループ0

○ 鶴岡 信治

濃度変換処理システムの概要

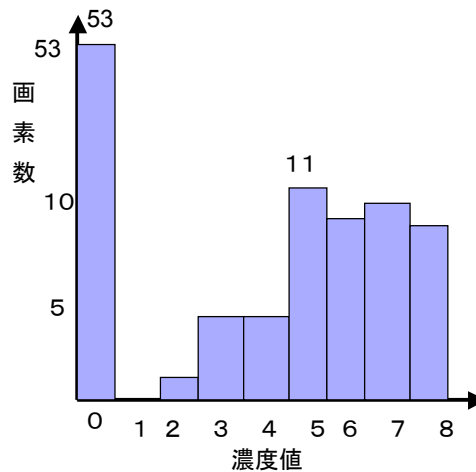
メニュー	ファイル (処理内容)	処理2 (処理内容)
メニュー 名 項目	開く (ディスクにある 画像ファイルをロー ド)	ヒストグラム (ロードした画像に対し てヒストグラムを作成し, pictureBox3 に表示)
	コピー2 (pictureBox1の 画像を pictureBox2に コピー)	線形濃度変換 (ロードした画像を対象 に濃度値を線形変換し, pictureBox2に 表示)
	コピー3 (pictureBox1の 画像を pictureBox2に コピー)	非線形濃度変換 (ロードした画像に対 象に濃度値を非線形変換し, pictureBox2に表示)
		コントラスト改善 (ロードした画像を 対象にコントラストを改善し, pictureBox2に表示)
		ヒストグラム平坦化 (ロードした画像 を対象に濃度値をヒストグラム平坦 化により非線形変換し, pictureBox2 に表示)

ヒストグラムとは？

画像全体に含まれる同じ濃度値の画素数をグラフ化

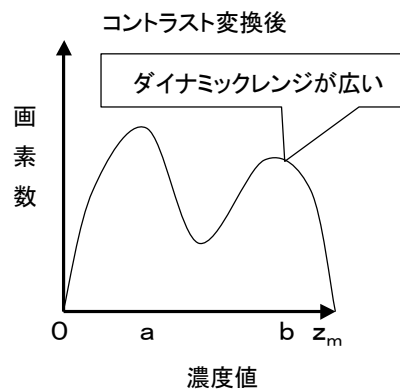
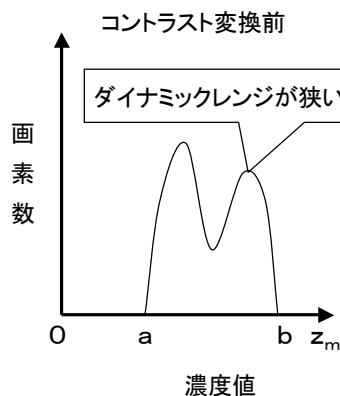
				2	5	3			
			3	5	8	5			
		4	5	7	8	7	5		
	3	7					5	7	
	6	8					5	6	
	3	6	7	5	4	7	8	5	
	5	6	8	8	7	6	8	5	
	4	7					5	7	
	6	6					6	7	
	4	5					3	4	

空白:濃度値0の画素

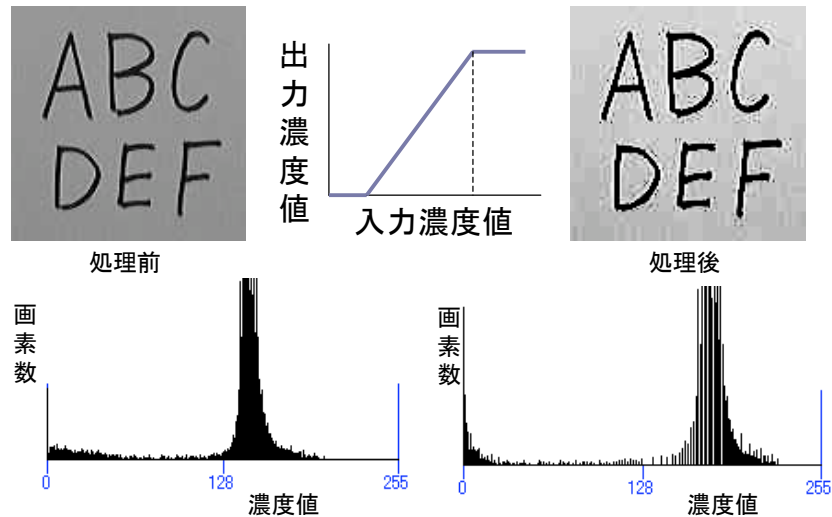


コントラストの改善

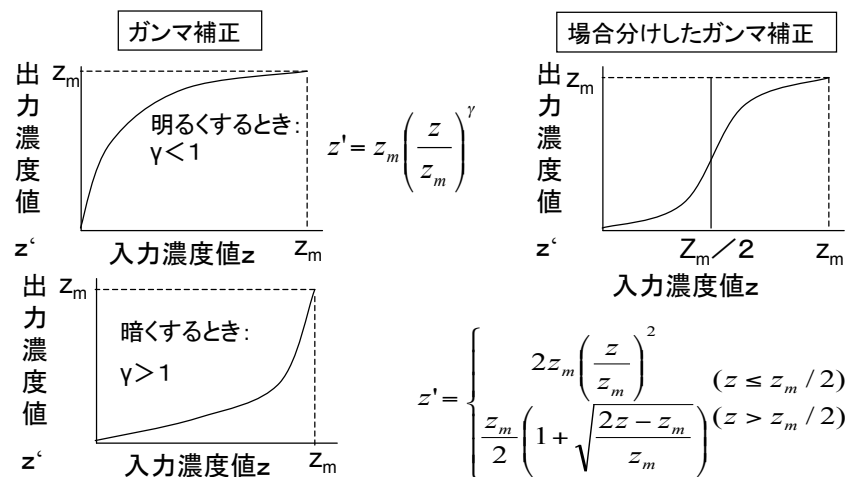
コントラスト:画像の明るい部分と暗い部分の明るさの比



コントラストの改善（線形濃度変換）



コントラストの改善 （非線形な濃度変換：ガンマ補正）



作成したプログラムの重要な部分

非線形濃度変換プログラム(入力画像: pb1構造体、出力: pb2に画像表示)

```
private void P2_NonLinear_Click(object sender, System.EventArgs e)
{
    int i, j, nx, ny;
    int gray;
    Color col;
    nx = pb1.Width;    /* 画像の幅 */
    ny = pb1.Height;   /* 画像の高さ */
    Bitmap bmp = new Bitmap(pb1.Image); /*ビットマップ構造体*/
    for(j = 0; j < ny; j++)          /* 各画素についての処理 */
        for(i = 0; i < nx; i++)
        {
            col = bmp.GetPixel(i,j); /* 各画素をcolor構造体 */
            gray = col.R;             /* 濃度値に赤を使用 */
        }
}
```

コントラストの改善処理

```
if(gray <= GMAX/2) /* 場合分けしたガンマ補正*/
    gray = (int)(2*GMAX * ((double)gray / (double)GMAX)*
                ((double)gray / (double)GMAX)); /* 比の2乗を計算 */
else
    gray = (int)(GMAX/2.0*(1+Math.Sqrt((2.0*(double)gray-
                (double)GMAX)/(double)GMAX))); /* ルートを計算 */

if( gray > GMAX ) gray = GMAX; /* grayをGMAX以下にする */
if( gray < 0 ) gray = 0;      /* grayを0以上にする */
bmp.SetPixel(i, j, Color.FromArgb(gray, gray, gray));
/* 各画素の濃度値をビットマップ画像に代入 */
} /* 画像の各画素ごとの処理の終わり */
pb2.Image = bmp; /* bmp画像をpictureBox2に代入 */
}
```

処理結果(線形濃度変換)



処理結果(非線形濃度変換)



感想

- (1) 数式で与えられた処理をプログラムで実現するには、かなり多くの知識と理解力が必要であることがわかった。
- (2) プログラミングは理論ではなくて、実践が重要であることがわかった。
- (3) プログラミングは難しいが、動いた時には感動した。

今後学習したい内容

- キーボード入力が大変なので、キーボード入力をなくすために、文字認識のアルゴリズムを研究したいと思った。
- 自動車の自動走行処理をカメラ画像を使用して行ってみたいと思った。